

PI Manager Catalog & Meta-Url

PI Manager uses ocpn-plugins.xml to provide the required metadata and URLs for installation of plugins.

Ocpn-plugins.xml and associated metadata files are located in <https://github.com/OpenCPN/plugins/>

Batch files and scripts are available to help create this file ocpn-plugins.xml is created from the individual plugin metadata files located in the data directory.

Please review the [Readme](#) file.

In summary

1. ocpn-plugins.xml is maintained within the project <https://github.com/OpenCPN/plugins>
2. Users can download updated versions directly from github (=this project <https://github.com/OpenCPN/plugins>).
3. Devs can clone this repo and create their own variant which is accessible from the GU using a custom URL.
4. Devs can issue PR:s against this repo to have their updates incorporated in the main ocpn-plugins.xml.
5. Dave has access to build the ocpn-plugins.xml in the master branch of the repository.

The scripts require Python

Add Metadata.xml files to the Beta Branch

Once you have build and deployed the 10-12 build environments to your Cloudsmith Repository, you should check the files in the repository and files, and then get the xml's into the Beta Catalog. For Example: <https://cloudsmith.io/~mauro-calvi/repos/squiddio-manual/packages/?page=1> Download each of the recent xml files to a local directory, using the green button at the end and selecting "Save Link".

Then from your local github directory,

```
git clone https://github.com/OpenCPN/plugins.git plugins
```

You want to clone this. <https://github.com/OpenCPN/plugins> Then checkout the Beta branch.

```
git checkout -b Beta
git fetch
```

Then copy your metada files (after looking at some of them for errors) to the metadata directory. Then, reading the README run the python script to join the metadata files into one file call ocpn_plugin.xml Then run the URL check on the file to make sure all urls are good.

Then you can make a PR to the Beta branch to Dave (bdbcat) at OpenCPN/plugins

Create a Meta-url on Alpha Branch

To enable the Alpha Channel you must make an entry in the ini / config file Also add how to turn on this feature by using the expert key

```
[PlugIns]
catalogexpert=1
```

META-URL's are only accepted on the github.com/opencpn/plugins/Alpha Branch.

1. From your local Github repository
`git clone https://github.com/[GitUser]/OpenCPN/plugins.git`
2. Then to go to the Alpha branch.
`git checkout Alpha`
3. From the Alpha branch create your own new corresponding branch which will contain your testing metadata.xml's.
`git checkout -b [YourName]-alpha`
4. Now go back to the Alpha branch to create your own META-URL xml file.
`git checkout Alpha`
5. Create and add your own Meta-url xml file in
[GitUser]/plugins/Alpha/metadata alongside the other Plugin Dev xml files.
6. This file will have a META-URL redirect to your new private branch. Example follows:
7. In this case the path & file is
[GitUser]/plugins/Alpha/metadata/meta-url-alpha-seandepagnier-rgleason.xml
8. The text in the xml file is:

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin version="1">
  <name> Sean Depagnier Alpha Plugins meta-url </name>
  <version> 1.0 </version>
  <summary> meta-url to Sean's Alpha Plugins by rgleason </summary>
  <meta-url>
https://raw.githubusercontent.com/rgleason/plugins/rg-alpha/ocpn-plugins.xml
  </meta-url>
</plugin>
```

1. Check the new file for format, accuracy and complete tags.
2. Next run the python tool to build a new ocpn-plugins.xml by executing at the command prompt from the /plugins/ directory
`python tools/ocpn-metadata generate -force -userdir metadata -destfile ocpn-plugins.xml`
3. See the README. This command collects all the files into one file. You will need to have python configured on your computer.
4. Then check that Alpha branch has a good ocpn-plugins.xml containing the xml data needed to direct to the cloud sources, including yours and other plugin meta-url redirects.

5. Now git push your changes to the Alpha branch up to your remote repository, and then push it to 'opencpn/plugins/Alpha'.
6. Wait for your PR to be accepted.
7. Your new private branch `[[github user]/plugins branch:[YourName]-alpha`
8. Now needs to be populated with metadata xmls from your Cloudsmith directory. In my case the branch is
`github.com/rgleason/OpenCPN/plugins/rg-alpha`
9. Copy Cloudsmith metadata xml files for plugins into
`github.com/rgleason/OpenCPN/plugins/rg-alpha/metadata`
10. Check those xml for format and complete tags.
11. Then collect the metadata xmls into a new `ocpn-plugins.xml` file with
`python tools/ocpn-metadata generate -force -userdir metadata -destfile ocpn-plugins.xml`
12. Note order of xml data may change, causing it to appear that the PR has more changes than it actually does.
13. Then push the new files in that branch up to your remote, so that there are some metadata.xml files for your new META-URL in the Alpha branch to link to, so that you can test some plugins in the current Developer's version of Opencpn.

[#19 Pull Request from opencpn/plugins/rgleason:rg-alpha to opencpn/plugins:Alpha](#)

Also here is another example with [squiddio](#)

From:

<https://opencpn.org/wiki/dokuwiki/> - **OpenCPN Manuals**

Permanent link:

https://opencpn.org/wiki/dokuwiki/doku.php?id=opencpn:developer_manual:pi_installer_opencpn_plugins

Last update: **2021/07/03 19:09**

